

1. Introduction

The purpose of this test plan is to ensure that the Order module of the e-commerce platform functions as expected, meets business requirements, and is prepared for future integrations.

2. Scope

This test plan covers functional and non-functional testing for the Order module, ensuring:

Users can add valid items to an order.

The system correctly calculates the total order cost.

Invalid items trigger appropriate error handling.

Performance and scalability requirements are met.

3. Test Objectives

Validate that orders are created, modified, and displayed correctly.

Ensure data integrity when adding/removing items.

Check error handling for invalid inputs.

Assess performance under load.

Verify future scalability aspects such as API integrations.

4. Test Approach

Both manual and automated testing will be employed:

Manual Testing: Functional Testing, boundary value analysis, and negative testing.

Automated Testing: Functional Testing, Regression testing, and performance testing.

5. Test Environment

Hardware: Standard user machine configurations.

Software: Test environment matching production setup.

Tools: Selenium for automation, JMeter for performance testing.

Browser Compatibility: Chrome, Firefox, Safari, Edge.

Device Compatibility: Desktop, tablet, and mobile devices.

6. Test Execution Plan

Phase 1: Manual functional testing.

Phase 2: Automated regression testing.

Phase 3: Performance and scalability testing.

Phase 4: Integration testing with future APIs.

7. Defect Tracking & Reporting

All defects will be logged in JIRA.

Severity and priority assigned based on impact.

Regular test status updates provided to stakeholders.

8. Exit Criteria

95% of test cases pass.

No critical defects remain unresolved.

Performance and scalability criteria met.

Cross-browser and device compatibility confirmed.

9. Risks & Mitigation

Risk	Mitigation
Performance degradation under high load	Optimize database queries, cache frequently used data.
Future API changes may break order functionality	Modular code design, thorough integration testing.
Database failure affecting orders	Implement backup and recovery mechanisms.
Multi-device UI inconsistency	Responsive design testing on various screen sizes

